Question 1. 5 points

Prove or disprove that:

(a) $[1 \text{ point}] n(n+1)/2 \in O(n^2)$ Correct $\frac{(n(n+1))}{2} = \frac{n^2 + n}{2} = \frac{n^2}{2} + \frac{n}{2} \leq N^2 \quad \forall \quad n \geq 1$

by using play = C = 1 no = 1

(b) [1 point] $n(n+1)/2 \in \Omega(n^2)$ by using polynomial $= \frac{n^2}{2} + \frac{n}{2} \stackrel{?}{=} \frac{1}{4} n^2$ thereon $n_0 = \left| \frac{\vec{k}_0 \cdot \vec{k}_0}{\vec{k}_0 - C} \right| C = \frac{1}{4}$ $\frac{n^2}{2} + \frac{n}{2} \stackrel{?}{=} \frac{1}{4} n^2 \quad \forall n \ge 2$ $C = \frac{1}{4} \quad n_0 = 2 \quad \text{correct}$

(c) [1 point] $n(n+1)/2 \in \Theta(n^2)$

from Previous Q = b we can see that $C_1 = \frac{1}{4} C_2 = 1 \max(h_a(1,2)) = 2$

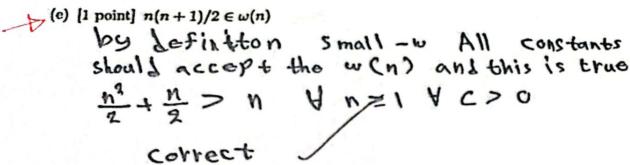
Correct

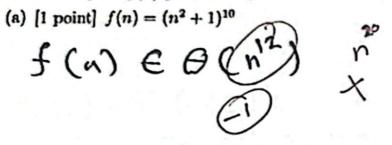
(d) [1 point] $n(n+1)/2 \in o(n^2)$ Uh Correct

by Lefintin of small the All constants C>, should accept a (n2)

 $\frac{n^2}{2} + \frac{n}{2} < n^2$ when c=1 n=1 $1 \neq 1$ So $\frac{n \cdot c \cdot n \cdot e \times 1}{2} \notin O(n^2)$

Page 2





(b) [1 point]
$$f(n) = 2n \log(n+2)^2 + (n+2)^2 \log(\frac{n}{2})$$

 $f(n) = 4n \log (n+2) + (h^2 + 4n + 4) (\log n - \log 2)$
 $f(n) = 4n \log (n+2) + h^2 \log n + 4n \log n + 4 \log n$
 $f(x) \in G(n^2 \log n)$

(c) [1 point]
$$f(n) = \sqrt{10n^2 + 7n + 3}$$

 $f(n) \in G(n)$

(d) [1 point]
$$f(n) = \log^2(n) \cdot \log(n^2)$$

 $f(n) \in \bigcap (\log^3(n))$

$$\begin{cases} T(n) = 3T(n/5) + n & \text{for } n > 1 \\ T(1) = 0 \end{cases}$$

$$T(n) = 3T(n/6) + n$$

 $T(n) = 3[3T(\frac{n}{5}) + \frac{n}{5}] + n$
 $T(n) = 3^{2}T(\frac{n}{5}) + \frac{3}{5}n + n$
 $= 3^{2}[3T(\frac{n}{5}) + \frac{n}{5}] + \frac{3}{5}n + n$
 $= 3^{2}T(\frac{n}{5}) + (\frac{3}{5})^{2}n + (\frac{3}{5})^{n}n + n$

aster·k step '
$$\tau(n) = 3^k \tau \left(\frac{n}{5^k}\right) + \sum_{j=0}^{k-1} \left(\frac{3}{5}\right)^j n$$

$$= 3^k \tau \left(\frac{n}{5^k}\right) + n \left(\frac{1 - \left(\frac{3}{5}\right)^k}{1 - \frac{3}{5}}\right)^k$$

$$= 3^k \tau \left(\frac{n}{5^k}\right) + n \left(\frac{1 - \left(\frac{3}{5}\right)^k}{1 - \frac{3}{5}}\right)^k$$

$$= 3^k \tau \left(\frac{n}{5^k}\right) + \frac{5}{2}n \left(1 + \left(\frac{3}{5}\right)^k\right)$$

$$\leq \frac{5}{2}n \in \mathcal{O}(n)$$

Page 4

```
Algorithm 1 Unknown(A[0..n-1])
                                             ▶ Input: An array A[0..n-1] of n real numbers
 1: minval ← A[0] -> \
2: maxval ← A[0] → 1
3: for i \leftarrow 1, n-1 do —
       if A[i] < minval then \longrightarrow h - 1
                                                                   f(n) = 5n-1
          minval \leftarrow A[i] \longrightarrow n-1
5:
       end if
6:
       if A[i] > maxval then \longrightarrow n - 1
7:
          maxval \leftarrow A[i] \longrightarrow N - 1
8:
 9:
       end if
10: end for
11: return maxval - minval ->
```

(a) [1 point] What does this algorithm compute?

this algorithm ditermains the max value in the array and ninimum value of the array and return the defence between the max value max value and return the defence between the max value of max value of the max valu

(b) [1 point] What is its basic operation?

the comparison in step 4
and step 7

(c)	[2 points] Give the best-case and worst-case time complexities of this algorithm in
	asymptotic notation. here the best-case
	and the worst-case are
	thesame
	f(n) = 5n-1 & 5n \ \ n = n
	C=5 ho=1
	and a comment of

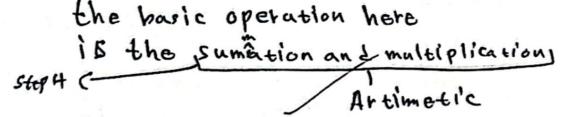
Algorithm 2 $S(n) \triangleright$ Input: A positive integer $n \triangleright$ Output: the sum of the first n squares

1: if n = 1 then \longrightarrow \(
2: \text{return } 1 \)

3: else

4: \text{return } $S(n-1) + n * n \longrightarrow \top (n-1) + 2$ 5: end if \bigwedge \(
\Lambda

(a) [1/2 point] What is the algorithm's basic operation?



K= N-1

(b) [2 points] Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed.

after k step

$$T(n) = T(n-k) + 2k$$

 $T(n) = 0 + 2(n-1) = EO(n)$
 $2n-2$

(c) [2 points] Give a pseudocode of a non-recursive version of this algorithm.

(d) [1/2 point] What is the worst-case time complexity of the algorithm in (c)?

Page 7

(e) [1 point] What is the suitable algorithm for computing S(n) (recursive or nonrecursive)? Justify your answer. bothe algorithms give us same time complixing but the recursive one stre take more space complixed than recarsiva (a) [3 points] Describe an algorithm that takes a list of n positive integers and finds the location of the last even integer in the list. It returns 0 if there are no even integers in the list. we can set the last element in the list and put it in a variable and Check the value at index m In to recursive or non recursive as 5- 1 loop and the loop stop when the condition is true and return the index in the array Linear search From Rast (b) [1 point] Give the best-case and worst-case time complexities of your algorithm in asymptotic notation. Show all details. the best-case is when the lost index is even = E (1) × 1/2 /

or there is no even in the list Mringu N

 $\longrightarrow \in \bigcirc$ (n)

Alg lusteren (A[1-n]) For I = n down to 1 If (A[I] mod2 = 0) peturn I y votory o Another solution Alg lasteren (A[1.n]) while (I>o and AIF]moch + 8) エ=エ-1 If (I>0) Leturn I 2 Else potura o

CSC 311 - Spring 2020

Assignment #1

Due: 8 February

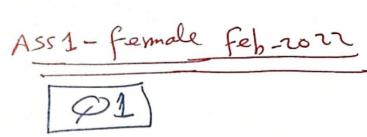
Question 4

Prove/disprove each of the following by finding appropriate values for c and no.

a)
$$n(n+1)/2 = O(n^2)$$
 // ignore negative
b) $2n^3 - 10n^2 + 2 = \Omega(n^3)$.

Question 5

For each of the following functions, indicate the class $\Theta(g(n))$ the function belongs to. (Use the simplest g(n) possible in your answers.) Prove your assertions.



it computes the sum of squares of first n

 $5(n) = 1^2 + 2^2 + 3^2 + \dots + n^2 = \sum_{i=1}^{n} i^2$

1 The Basic operation وهي العملية سين الي يتم تكروها مع كل لهذه is Multiplication

Multiplication and addition

- عدد وات تكرار او تنفيذ العليم الأسعي هرعدد مراست تکرار اللوب = ۱۱
 - 0(n)
 - سنافاعدة لجوع ربيعات إلار $5(n) = 1^2 + 2^2 + \cdots + n^2 = \sum_{i=1}^{n} = \frac{n(i+1)(2n+1)}{2n+1}$

Data: Anonnegative integern S = n * (n+1) * (2*n+1)/6

beturn 5

QB

Consider the following algorithm.

```
Algorithm Secret(A[0..n-1])

//Input: An array A[0..n-1] of n real numbers minval \leftarrow A[0]; maxval \leftarrow A[0]

for i \leftarrow 1 to n-1 do if A[i] < minval minval \leftarrow A[i] if A[i] > maxval minval \leftarrow A[i] return maxval \leftarrow A[i]
```

solution

 a. Computes the range, i.e., the difference between the array's largest and smallest elements.

b. An element comparison.

c.
$$C(n) = \sum_{i=1}^{n-1} 2 = 2(n-1)$$
.

d. $\Theta(n)$.

e. An obvious improvement for some inputs (but not for the worst case) is to replace the two if-statements by the following one:

 $\text{if } A[i] < minval \ minval \leftarrow A[i] \\$

else if $A[i] > maxval \ maxval \leftarrow A[i]$.

Another improvement, both more subtle and substantial, is based on the observation that it is more efficient to update the minimum and maximum values seen so far not for each element but for a pair of two consecutive elements. If two such elements are compared with each other first, the updates will require only two more comparisons for the total of three comparisons per pair. Note that the same improvement can be obtained by a divide-and-conquer algorithm.

Rotale Mines

$$\begin{array}{c|c}
\hline
 & V_{10n^{2}+7n+3} \\
\hline
 & V_{20n^{2}} \\
\hline
 & V_{20} \cdot V_{n2} \\
\hline
 & V_{20} \cdot V$$